Noname manuscript No.
(will be inserted by the editor)

Geometry-based Symbolic Approximation for Fast Sequence Matching on Manifolds

Rushil Anirudh · Pavan Turaga

Received: date / Accepted: date

Abstract In this paper, we consider the problem of fast and efficient indexing techniques for sequences evolving in non-Euclidean spaces. This problem has several applications in the areas of human activity analysis, where there is a need to perform fast search, and recognition in very high dimensional spaces. The problem is made more challenging when representations such as landmarks, contours, and human skeletons etc. are naturally studied in a non-Euclidean setting where even simple operations are much more computationally intensive than their Euclidean counterparts. We propose a geometry and data adaptive symbolic framework that is shown to enable the deployment of fast and accurate algorithms for activity recognition, dynamic texture recognition, motif discovery. Toward this end, we present generalizations of key concepts of piece-wise aggregation and symbolic approximation for the case of non-Euclidean manifolds. We show that one can replace expensive geodesic computations with much faster symbolic computations with little loss of accuracy in activity recognition and discovery applications. The framework is general enough to work across both Euclidean and non-Euclidean spaces, depending on appropriate feature representations without compromising on the ultra-low bandwidth, high speed and high accuracy. The proposed methods are ideally suited for real-time systems and low complexity scenarios.

Keywords manifold trajectories, sequence indexing, activity recognition, differential geometry, data mining

Rushil Anirudh, Pavan Turaga
School of Electrical, Computer and Energy Engineering
School of Arts, Media and Engineering
Arizona State University, Tempe AZ
E-mail: ranirudh@asu.edu

1 Introduction

In this paper we consider the problem of fast comparison of sequences of structured visual representations, which have non-Euclidean geometric properties. Examples of such structured representations include shapes [Kendall, 1984, Srivastava et al., 2011], optical flow [Chaudhry et al., 2009], covariance matrices [Tuzel et al., 2006] where underlying distance metrics are highly involved and even simple statistical operations are usually iterative.

Utilizing Riemannian geometric concepts have resulted in many advances in understanding complex representations. For example, features such as contours[Joshi et al., 2007], skeletons [Vemulapalli et al., 2014], the space of $d \times d$ covariance matrices or tensors which appear both in medical imaging [Pennec et al., 2006] as well as texture analysis [Tuzel et al., 2006] etc., have proven effective in image analysis. In video analysis, techniques have included temporal information using Riemannian properties such as, video modeling by linear dynamic systems [Turaga et al., 2011], and tensor decomposition [Lui et al., 2010] etc. Long-term complex activities are often modeled as time-varying linear dynamical systems [Turaga and Chellappa, 2009], which can be interpreted as a sequence of points on a Grassmann manifold, motivating application for the problem of indexing of manifold sequences.

For these manifolds, standard notions of distance, statistics, quantization etc. need modification to account for the non-linearity of the underlying space. As a result, basic computations such as geodesic distance, finding the sample mean etc are highly involved in terms of computational complexity, and often result in long iterative procedures further increasing the computational load making them impractical. To address this issue, in this paper we propose a geometry-

based symbolic approximation framework, as a result of which low-bandwidth transmission and accurate real-time analysis for recognition or searching through sequential data become fairly straightforward.

We propose a framework that generalizes a popular indexing technique used to mine and search for vector space time series data known as Symbolic Aggregate Approximation (SAX) [Lin et al., 2003] to Riemannian manifolds. To the best of our knowledge, we are the first to propose such an indexing scheme for manifold sequences. The main idea is to replace manifold sequences with abstract *symbols* or *prototypes*, that can be learned offline. Symbolic approximation is a combination of discretization and quantization on manifold spaces, which allows us to approximate distance metrics between sequences in a quick and efficient manner. Another advantage is extremely fast searching that is possible because the searching is limited to the symbolic space. Further, to enable efficient searching techniques, we develop prototypes or symbols which divide the space into equi-probable region by proposing the first manifold generalization of a conscience based competitive learning algorithm [Desieno, 1988]. Using the proposed prototypes, we demonstrate that signals or sequences on manifolds can be approximated effectively such that the resulting metric remains close to the metric on the original feature space, thereby guaranteeing accurate recognition and search. While this framework is applicable to general high-dimensional feature sequences, we demonstrate its utility on the in a few common video-analysis problems such as activity analysis and dynamic texture modeling. Generally speaking, the ideal symbolic representation is expected to have the following key properties: (1) Be able to model the data accurately with a low approximation error (2) Robust learning framework that is invariant to noise and outliers and, (3) A symbolic representation should enable the efficient use of existing data structures and algorithms developed for string searching.

We summarize our contributions next.

Contributions:

1. We present a geometry based data-adaptive strategy for indexing time series evolving on non-Euclidean spaces. We demonstrate the effectiveness on three manifolds namely the hypersphere, the Grassmann and the product space of $SE(3) \times \cdots \times SE(3)$.
2. We propose the first generalization of competitive learning algorithms to Riemannian manifolds for this task, such that they are able learn prototypes which enable efficient searching.
3. The resulting framework allows the comparison between two manifold sequences at speeds nearly $100\times$ faster than geodesic based comparisons.
4. Applications in activity recognition and discovery show that the speed up can be achieved with minimal loss of accuracy as compared to the original features.

Organization: In Sec 2, we discuss works related to indexing on non-Euclidean and Euclidean spaces. Next, Sec 3 introduces the manifolds used in this paper namely - Grassmann, Hypersphere, and the product space of $SE(3)$, including their geometric properties. Sec 4 introduces the extension of SAX to Riemannian manifolds, which includes the generalization of conscience based competitive learning in algorithm 1. Sec 5 discusses the application of string-based algorithms to speedup search and discovery of manifold sequences, applied to human activities. Finally, Sec 6 discusses the experiments on different manifold valued features on different activity datasets. We conclude the paper and disucss possible extensions and generalizations in Sec 7.

2 Related Work

Indexing static points on non-Euclidean spaces Not surprisingly, many standard approaches for sequence modeling and indexing which are designed for vector-spaces need significant generalization to enable application to these non-Euclidean spaces. Indexing of static data on manifolds has been addressed recently with hashing based approaches [Chaudhry and Ivanov, 2010]. For data points lying on the space of Symmetric Positive Definite (SPD) matrices, [Harandi et al., 2014] address a dimensionality reduction technique that is geometry aware. Our interest lies in indexing sequences directly instead of individual points. Signal approximation for manifolds using wavelets [Rahman et al., 2005] is a related technique. However, it is non-adaptive to the data and requires observing the entire signal before it can be approximated, while the proposed framework allows for easy real time implementation once the symbols are learned. Recent work also dealt with modeling human activity as a manifold valued random process [Yi et al., 2012] where the proposed techniques are theoretically and computationally involved due to the requirement of second-order properties such as parallel transports. Another related line of work in recent years has been advances in Riemannian metrics for sequences on manifolds [Srivastava et al., 2011]. These approaches consider a sequence as an equivalent vector-field on the manifold. A distance function is imposed on such vector-fields in a square-root elastic framework. This is applied to the special case of curves in $2D$, nD, and non-Euclidean spaces [Srivastava et al., 2011, Joshi et al., 2007, Su et al., 2014]. While such a distance function could be utilized for the purposes of indexing and approximation of se-

quences, it is offset by the computational load required in computing the distance function for long sequences.

Computationally efficient representations of images and video In past decade, there has been a significant progress in speeding up retrieval and indexing images [Chum et al., 2009] which are efficient in accurately retrieving similar images from very large datasets. There have also been extensions to video retrieval [Revaud et al., 2013] from very large databases. These techniques have made it possible to search accurately through large image and video data bases, but most methods are for high dimensional Euclidean static points instead of time series. Perhaps [Revaud et al., 2013] is the most related to our work in that they address the efficient retrieval and indexing of Euclidean time series, however, the generalization to manifold sequences is unclear.

Euclidean time series indexing A successful approach to tackle the problem of fast indexing of *scalar* sequences has been to discretize and quantize the sequence in a way such that the obtained symbolic form contains most of the information of the original sequence, yet enabling much faster computations. This class of approaches are broadly termed as **S**ymbolic **A**ggregate Approx**i**mation (SAX) [Lin et al., 2003]. Several problems of indexing and motif discovery from time series have been addressed using this framework [Lin et al., 2003, Mueen et al., 2009], however the extension from $1D$ to multidimensional and non Euclidean spaces is not trivial. Multidimensional extensions to SAX have also been proposed such as [Vahdatpour et al., 2009], but these are trivial extensions which perform SAX on every dimension individually without considering the geometry of the ambient space.

Further, for manifolds such as the Grassmannian or the function-space of closed curves, there is no natural embedding into a vector space, thus motivating the need for a geometry-based intrinsic approach [Spivak, 1999, Srivastava et al., 2011]. We show that this class of approaches can be generalized to take into account geometry of the feature space resulting in several appealing characteristics for manifold-valued time-series, as they enable us to replace highly non-linear distance function computations with much faster and simpler symbolic distance computations.

Efficient string searching The biggest advantage of using the proposed indexing method is the the representation of complex feature types using abstract symbols, that are learned offline. This enables the use of string searching algorithms, allowing one to search through very high dimensional, non-linear spaces with a $O(m+n)$ complexity or better, where m and n are the length of a query and string respectively. A known result in data mining is that the computational complexity can be further reduced to $O(m + n(log_{|\Sigma|}m)/m)$, for an alphabet of size Σ, when the letters of the alphabet are independent and equiprobable [Allauzen and Raffinot, 2000]. Other lower bounds have been proposed when letters are equiprobable [Yao, 1979], and it is known the height of suffix trees is optimized with equiprobable letters [Devroye et al., 1992]. The vector space SAX [Lin et al., 2003] proposed to generate symbols by partitioning the Gaussian distribution into bins of equal probability. However, it is not trivial to partition the data space into equiprobable regions on manifolds hence we use a conscience based competitive learning algorithm to learn the symbol alphabet.

3 Mathematical Preliminaries

In this section we will outline the geometric properties of the manifolds considered in this work, namely the Grassmann, hyper-sphere and the space of $SE(3) \times \ldots SE(3)$. For an overview on Riemannian geometry and topology, we refer the readers to useful resources on the topic [Absil et al., 2004, Boothby, 2003]. Next we describe the different features and their respective geometric spaces.

Landmarks on the Silhouette: We represent a shape as a $m \times 2$ matrix $L = [(x_1, y_1); (x_2, y_2); \ldots; (x_m, y_m)]$, of the set of m landmarks of the zero-centered shape. The *affine shape space* [Goodall and Mardia, 1999] is useful to remove the effects of small variations in camera location or small changes in the pose of the subject. Affine transforms of the base shape L_{base} can be expressed as $L_{affine}(A) = L_{base} * A^T$, and this multiplication by a full-rank matrix on the right preserves the column-space of the matrix L_{base}. Thus, the 2D subspace of $\mathbb{R}^m$ spanned by the columns of the matrix L_{base} is an *affine-invariant* representation of the shape. i.e. $span(L_{base})$ is invariant to affine transforms of the shape. Subspaces such as these can be identified as points on a Grassmann manifold [Turaga et al., 2011].

A given d-dimensional subspace of $\mathbb{R}^m$, $\mathcal{Y}$ can be associated with a idempotent rank-d projection matrix $P = YY^T$, where Y is a $m \times d$ orthonormal matrix such as $span(Y) = \mathcal{Y}$. The space of $m \times m$ projectors of rank d, denoted by $\mathbb{P}_{m,d}$ can be embedded into the set of all $m \times m$ matrices - $\mathbb{R}^{m \times m}$- which is a vector space. Using the embedding $\Pi : \mathbb{R}^{m \times m} \rightarrow \mathbb{P}_{m,d}$ we can define a distance function on the manifold using the metric inherited from $\mathbb{R}^{m \times m}$.

$$d^2(P_1, P_2) = tr(P_1 - P_2)^T(P_1 - P_2) \tag{1}$$

The projection $\Pi : \mathbb{R}^{m \times m} \rightarrow \mathbb{P}_{m,d}$ is given by:

$$\Pi(M) = UU^T \tag{2}$$

where $M = USV^T$ is the d-rank SVD of M. (6)

Given a set of sample points on the Grassmann manifold represented uniquely by projectors $\{P_1, P_2, ...P_N\}$, we can compute the extrinsic mean [Turaga et al., 2010] by first computing the mean of the P_i's and then projecting it to the manifold as follows :

$$\mu_{ext} = \Pi(P_{avg}), \text{where } P_{avg} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{3}$$

Histograms of Oriented Optical Flow (HOOF): As described in [Chaudhry et al., 2009], optical flow is a natural feature for motion sequences. Directions of Optical Flow vectors are computed for every frame, then binned according to their primary angle with the horizontal axis and weighted according to their magnitudes. Using magnitudes alone is susceptible to noise and can be very sensitive to scale. Thus all optical flow vectors, $v = [x, y]^T$ with direction $\theta = tan^{-1}(\frac{y}{x})$ in the range

$$-\frac{\pi}{2} + \pi\frac{b-1}{B} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B} \tag{4}$$

will contribute by $\sqrt{x^2+y^2}$ to the sum in bin b, $1 \leq b \leq B$, out of a total of B bins. Finally, the histogram is normalized to sum up to 1. Each frame is represented by one histogram and hence a sequence of histograms are used to describe an activity. The histograms h_t = $[h_{t;1}$, $\ldots$, $h_{t;B}]$ can be re-parameterized to the *square root representation* for histograms, $\sqrt{\mathbf{h_t}} = [\sqrt{h_{t;1}}, \ldots, \sqrt{h_{t;B}}]$ such that $\sum_{i=1}^{B}(\sqrt{h_{t;i}})^2 = 1$. The Riemannian metric between two points R_1 and R_2 on the hypersphere is $d(R_1, R_2) = cos^{-1}(R_1^T R_2)$. This projects every histogram onto the unit B-dimensional hypersphere or $\mathbb{S}^{B-1}$. From the differential geometry of the sphere, the exponential map is defined as [Srivastava et al., 2007]

$$\exp_{\psi_i}(v) = \cos(||v||_{\psi_i})\psi_i + \sin(||v||_{\psi_i})\frac{v}{||v||_{\psi_i}} \tag{5}$$

Where $v \in T_{\psi_i}(\Psi)$ is a tangent vector at ψ_i and $||v||_{\psi_i}$ = $\sqrt{\langle v, v\rangle_{\psi_i}} = (\int_0^T v(s)v(s)ds)^{\frac{1}{2}}$. In order to ensure that the exponential map is a bijective function, we restrict $||v||_{\psi_i} \in [0, \pi]$. The truncation of the domain of the the exponential map is made in accordance to the injectivity radius, which is the largest radius for which the exp map is a diffeomorphism. For the sphere, the injectivity radius is π. Points that lie beyond the injectivity radius have a shorter path connecting them to ψ_i, which determines their geodesic distance incorrectly. The logarithmic map from ψ_i to ψ_j is then given by

$$\overrightarrow{\psi_i\psi_j} = \log_{\psi_i}(\psi_j) = \frac{\mathbf{u}}{(\int_0^T \mathbf{u}(s)\,\mathbf{u}(s)ds)^{\frac{1}{2}}}\cos^{-1}\langle\psi_i, \psi_j\rangle,$$

with $\mathbf{u} = \psi_i - \langle\psi_i, \psi_j\rangle\,\psi_j$.

Lie Algebra Relative Pairs (LARP): Finally, we consider a skeletal representation proposed recently [Vemulapalli et al., 2014] which has been shown to be very effective for activity recognition on data obtained from depth sensors such as Microsoft Kinect. LARP represents every skeleton as a set of relative transformations between joints, where a transformation consists of a rotation and a translation and therefore lies on the Special Euclidean group $SE(3)$. Further every skeleton with N joints, is represented as a set of such transformations between $\binom{N-1}{2}$ relative pairs, therefore the final feature is represented as a point on a product space of $SE(3) \times \cdots \times SE(3)$.

The special Euclidean group, denoted by $SE(3)$ is a Lie group, containing the set of all 4×4 matrices of the form

$$P(R, \overrightarrow{d}) = \begin{bmatrix} R & \overrightarrow{d} \\ 0 & 1 \end{bmatrix}, \tag{7}$$

where R denotes the rotation matrix, which is a point on the special orthogonal group $SO(3)$ and $\overrightarrow{d}$ denotes the translation vector, which lies in $\mathbb{R}^3$. The 4×4 identity matrix I_4 is an element of $SE(3)$ and is the identity element of the group. The exponential map, which is defined as $\exp_{SE(3)}$: $\mathfrak{se}(3) \to SE(3)$ and the inverse exponential map, defined as $\log_{SE(3)} : SE(3) \to \mathfrak{se}(3)$ are used to traverse between the manifold and the tangent space respectively. The exponential and inverse exponential maps for $SE(3)$ are simply the matrix exponential and matrix logarithms respectively, from the identity element I_4. The tangent space at I_4 of a $SE(3)$ is called the Lie algebra of $SE(3)$, denoted by $\mathfrak{se}(3)$. It is a 6-dimensional space formed by matrices of the form:

$$B = \begin{bmatrix} U & \overrightarrow{w} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -u_3 & u_2 & w_1 \\ u_3 & 0 & -u1 & w_2 \\ -u_2 & u_1 & 0 & w_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8}$$

where U is a 3×3 skew-symmetric matrix and $\overrightarrow{w} \in \mathbb{R}^3$. An equivalent representation of B is $\text{vec}(B) = [u_1, u_2, u_3, w_1, w_2, w_3]$ which lies on $\mathbb{R}^6$.

These tools are trivially extended to the product space, for example the identity element of the product space is simply $(I_4, I_4, \ldots, I_4)$ and the Lie algebra is $\mathfrak{m} = \mathfrak{se}(3) \times \cdots \times \mathfrak{se}(3)$.

4 Symbolic approach for Manifold Sequences

In this section, we describe the proposed representation for manifold sequences which allows efficient algorithms to be deployed for a variety of tasks such as motif discovery, low-complexity activity recognition. We focus on the piece-wise aggregate and Symbolic approximation (PAA, SAX)[Chakrabarti et al., 2002,Lin et al., 2003] formulation, and present an intrinsic method to extend it to non Euclidean spaces like manifolds. Briefly, the PAA and SAX formulation consist of the following principal ideas - A given 1D scalar time-series is first divided into windows and the sequence in each window is represented by its mean value. This process is referred to as **piece-wise aggregation**. Then, a set of 'break-points' is chosen which correspond to dividing the range of the time-series into equi-probable bins. These break-points comprise the symbols using which we translate the time series into its symbolic form. For each window, the mean value is assigned to the closest symbol, this step is referred to as **symbolic approximation**. This representation has been shown to enable efficient solutions to scalar time-series indexing, retrieval, and analysis problems [Lin et al., 2003].

In the manifold case, to enable us to exploit the advantages offered by the symbolic representation of sequences, we need solutions to the following main problems - a) piece-wise aggregation: which can be achieved by appropriate definitions of the mean of a windowed sequence on a manifold, and b) symbolic approximation: which requires choosing a set of points that are able to represent the data well. Here, we discuss how to generalize these concepts for the case of manifolds.

4.1 Piece-wise aggregation

Given a sequence $\gamma(t) \in \mathcal{M}$, we define its piece-wise approximation in terms of local-averages in small time-windows. To do this, we first need a notion of a mean of points on a manifold. Given a set of points on a manifold, a commonly used definition of their mean is the Riemanian center of mass or the Frétchet mean [Karcher, 1977], which is defined as the point μ that minimizes the sum of squared-distance to all other points: $\mu = \arg\min x \in \mathcal{M} \sum_{i=1}^{N} d(x, x_i)^2$, where d is the geodesic distance on the manifold.

Computing the mean is not usually possible in a closed form, and is unique only for points that are close together [Karcher, 1977]. An iterative procedure is popularly used in estimation of means of points on manifolds [Pennec, 2006]. Since in local time windows, points are not very far away from each other, the algorithm always converges. Thus, given a manifold-valued time series $\gamma(t)$, and a window of length W, we compute the mean of the points in the window and this gives rise to the piece-wise aggregate approximation for manifold sequences. When we consider vectors in $\mathbb{R}^n$, this reduces to finding the standard mean of W n-dimensional vectors. The importance of this step is to reduce the number of points within the time series, a shorter sequence is computationally much faster to compute and store, but has a trade-off with increased error of approximation as shown in figure 3.

4.2 Symbolic approximation

As discussed above, one of the key-steps in performing symbolic approximation for manifold-valued time-series is to obtain a set of discrete symbols. An established theoretical result within the data mining literature is that the efficiency of string searching is optimized when the letters of the alphabet are equiprobable [Allauzen and Raffinot, 2000, Devroye et al., 1992]. The authors of SAX [Lin et al., 2003] emphasize on using equi-probable symbols because they achieve optimal results for fast searching and retrieval using suffix trees, hashing, and Markov models. However, standard clustering approaches do not necessarily result in equiprobable distributions of their centers [Zador, 1982, Kohonen, 1995,Ripley, 1996]. It is also known that when symbols are not equiprobable, there is a possibility of inducing a probabilistic bias in the process [Lin and Li, 2010]. We outline the methods to obtain symbols next.

4.2.1 Geometry aware K-means for learning symbols

While any clustering approach could be used for this step, we chose K-means because it is the most widely used clustering approach and its extension to non Euclidean spaces is well understood. For a set of points $D = (U_1, U_2, \ldots, U_n)$ we seek to estimate clusters $(C) = (C_1, C_2, \ldots, C_K)$ with centers $(\mu_1, \mu_2, \ldots, \mu_K)$ such that the sum of geodesic-distance squares, $\Sigma_{i=1}^{K} \Sigma_{U_j \in C_i} d^2(U_j, \mu_i)$ is minimized. Here $d^2(U_j, \mu_i) = |\exp_{\mu_i}^{-1}(U_j)|^2$, where $\exp^{-1}$ is the inverse exponential map as described in section 3.

4.2.2 Conscience based competitive learning on manifolds

To generate symbols or prototypes that divide the feature manifold into equiprobable regions, we extend ideas from Desieno's competitive learning mechanism [Desieno, 1988] to make it adaptive to the geometry of the space and generate equiprobable symbols. It has been observed that a 'conscience' based competitive learning approach does result in symbols that are much more equiprobable than those obtained from clustering approaches. However, the algorithm described in [Desieno, 1988] is devised only for vector-spaces. Here, we present a generalization of this approach

to account for non-Euclidean geometries. The tools that we build upon include computation of geodesic distances, exponential maps and inverse-exponential maps. These are known for many standard manifolds commonly occurring in computer vision applications.

The conscience mechanism starts with a set of initial symbols/prototypes. When an input data-point is presented, a competition is held to determine the symbol closest in distance to the input point. Here, we use the geodesic distance on the manifold for this task. Let us denote the current set of K symbols as $\{S_1, S_2, \ldots, S_K\}$, where each $S_i \in \mathcal{M}$. Let the input data point be denoted as $X \in \mathcal{M}$. The output y_i associated with the i^{th} symbol is described as

$$y_i = 1, \text{ if } d^2(S_i, X) \leq d^2(S_j, X), \forall j \neq i \quad (9)$$
$$y_i = 0, \text{otherwise}$$

where, $d()$ is the geodesic distance on the manifold. Since this version of competition does not keep track of the fraction of times each symbols wins, it is modified by means of a bias term to promote more equitable wins among the symbols. A bias b_i is introduced for each symbol based on the number of times it has won in the past. Let p_i denote the fraction of times symbol i wins the competition. This is updated after each competition as

$$p_i^{new} = p_i^{old} + B(y_i - p_i^{old}) \quad (10)$$

where $0 < B << 1$. The bias b_i for each symbol is computed as $b_i = C(\frac{1}{K} - p_i)$, where C is a scaling factor chosen to make the bias update significant enough to change the competition (see below). The modified competition is given by

$$z_i = 1, \text{ if } d^2(S_i, X) - b_i \leq d^2(S_j, X) - b_j, \forall j \neq i \quad (11)$$
$$z_i = 0, \text{otherwise.}$$

Finally, the winning symbol is adjusted by moving it partially towards the input data point. The key extension of this algorithm from vector space to non Euclidean spaces lies in this step. In the vector-space version this step is achieved by $S_i^{new} = S_i^{old} + \alpha((X) - S_i^{old})z_i$, but generalization of operations such as subtraction and multiplication to manifolds are not trivial. The partial movement of a symbol towards a data-point can be achieved by means of the exponential and inverse-exponential map as

$$S_i^{new} = \exp_{S_i^{old}}[\alpha \exp^{-1}_{S_i^{old}}(X) z_i]. \quad (12)$$

The proposed algorithm for conscience based equi-probable symbol learning is summarized in algorithm 1.

Next, we illustrate the strength of this approach in obtaining equiprobable symbols on manifolds. For this experiment we chose the UMD human activity dataset [Veeraraghavan and Chowdhury, 2006] and pre-processed it such that we obtain the outer contour of the human. A detailed discussion of the dataset, processing, choice of shape metrics etc. appears in the experiments section. Here, we performed clustering of the shapes into 5 clusters and used the centroids as symbols. We show the histograms of the symbols as obtained in fig 1. As can be seen, both K-means and affinity propagation result in symbols that are far from equiprobable. The proposed approach results in symbols which are much closer to a uniform distribution. The entropy defined as $-\Sigma_i p_i log_2(p_i)$, is shown for three different datasets in fig 2. It is seen that the algorithm converges quickly in all cases. Once the symbols are obtained, transforming the feature sequence to its symbolic form is performed using algorithm 2.

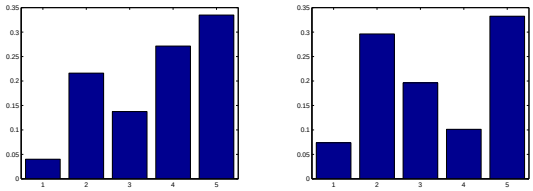
In practice, while K-means minimizes approximation error it does not have the favorable property of equiprobability, and competitive learning gives us symbols which are equally likely, while compromising on approximation error. In order to find a trade-off between the two, we use a hybrid approach that first uses K-means and then competitive learning from which equiprobable symbols can be obtained in a two stage process. In the first stage we cluster the data using K-means into a small number of clusters, this ensures most data points are adequately represented. Each of these clusters is further split into smaller, equiprobable *sub-clusters*

Algorithm 1 Equiprobable symbol generation on manifolds.

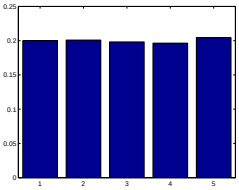
Input: Dataset $\{X_1, \ldots, X_n\} \in \mathcal{M}$. Initial set of symbols $\{S_1, \ldots, S_k\}$.
Parameters: Biases $b_i = 0$, learning rate α, win update factor B, conscience factor C.
while $iter \leq maxiter$ **do**
  for $j = 1 \rightarrow n$ **do**
    $\tilde{i} \leftarrow \min_i d^2(X_j, S_i) - b_i$
    $z_{\tilde{i}} = 1, z_i = 0, i \neq \tilde{i}$
    $S_i \leftarrow \exp_{S_i}[\alpha \exp^{-1}_{S_i}(X_j) z_i]$
    $p_i \leftarrow p_i + B(z_i - p_i)$
    $b_i \leftarrow C(1/k - p_i)$
  end for
end while

Algorithm 2 Symbolic Approximation for Feature Sequences in Euclidean & Non Euclidean Spaces.

Input: Feature sequence $\{\beta_1, \ldots, \beta_N\} \in \mathcal{M}$, Learned dictionary $\{D_1, \ldots, D_K\}$, Metric $d_{\mathcal{M}}$ defined on $\mathcal{M}$
Parameters: Size of aggregating window W ($<< N$),
Output: Symbolic approximation, **S**.
$M \leftarrow \lceil \frac{N}{W} \rceil$.
$n = 1$
for $m = 1 \rightarrow M$ **do**
  $\mathrm{A}_m \leftarrow$ **intrinsic mean**$\{\beta_n, \beta_{n+1} \ldots \beta_{n+W-1}\}$
  $\mathbf{S}(m) \leftarrow \underset{1 \leq j \leq K}{\text{argmin}} \; d_{\mathcal{M}}(A_m, D_j)$.
  $n = n + m \times W$
end for



(a) K-Means (b) Affinity Propagation



(c) Equi Probable Clustering

Fig. 1: Probability Density Functions of the labels generated using (a) K-Means clustering, (b) Affinity Propagation and (c) Equi-Probable Clustering are shown, the feature space in this case was the Grassmann manifold as described in the text. As seen above, equiprobable clustering assigns all clusters with almost equal probability.

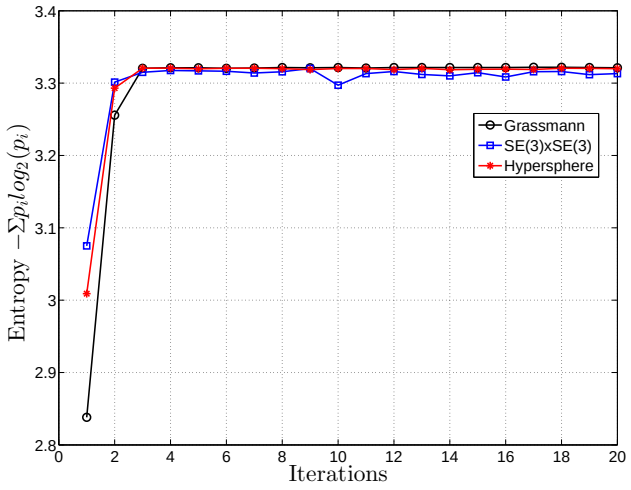


Fig. 2: Convergence for the algorithm 1 on different feature manifolds to obtain 10 symbols - Grassmannian (UMD), Hypersphere (Weizmann) and $SE(3) \times .. \times SE(3)$(UTKinect). Entropy is plotted as a measure of equiprobability, higher the better.

in the second stage using conscience learning. The number of clusters in the first stage is an empirical choice, we used values in the range of 5 to 10 for each data set. The number of sub-clusters in the second stage varies according to the probability of their parent cluster. For example, if p_s was the probability of the smallest cluster and we decide to split it into r smaller sub-clusters, then the i^{th} cluster with probability p_i would be split into $\lceil \frac{p_i}{p_s} \times r \rceil$ clusters. The parameter r indirectly controls the size of the final set of symbols, we used values of r in the range of 1 to 5. We chose these values to obtain a symbol set of size($\sim 40 - 50$). The training for symbols is expected to be computationally intensive, however this needs to be done only once and can be performed offline and does not affect the speed of comparisons during testing.

4.3 Limitations and special cases

Here, we discuss the limitations and some special cases of the proposed formulation. The overall approach assumes that a training set can be easily obtained from which we can extract the symbols for sequence approximation. In the 1D scalar case, this is not an issue, and one assumes that data distribution is a Gaussian, thus the choice of symbols can be obtained in closed-form without any training. If data is not Gaussian, a simple transformation/normalization of the data can be easily performed. In the manifold case, there is no simple generalization of this idea, and we are left with the option of finding symbols that are adapted for the given dataset.

For the special case of $\mathcal{M} = \mathbb{R}^n$, the approach boils down to familiar notions of piece-wise aggregation and symbolic approximation with the additional advantage of obtaining data-adaptive symbols, this ensures that the proposed approach is applicable even to the vast class of traditional features used in video analysis. For the case of manifolds implicitly specified using samples, we suggest the following approach. One can obtain an embedding of the data into a Euclidean space and apply the special case of the algorithm for $\mathcal{M} = \mathbb{R}^n$. The requirement for the embedding here is to preserve geodesic distances between local pairs of points, since we are only interested in ensuring that data in small windows of time are mapped to points that are close together. Any standard dimensionality reduction approach [Tenenbaum et al., 2000, Roweis and Saul, 2000] can be used for this task. However, recent advances have resulted in algorithms for estimating exponential and inverse exponential maps numerically from sampled data points [Lin and Zha, 2008]. This would make the proposed approach directly applicable for such cases, without significant modifications. Thus the proposed formalism is applicable to manifolds with known geometries as well as to those whose geometry needs to be estimated from data.

5 Speed up in sequence to sequence matching using symbols: applications in activity recognition and discovery

The applications considered in this paper are recognition and discovery of human activities. For recognition, a very commonly used approach involves storing labeled sequences for each activity, and performing recognition us-

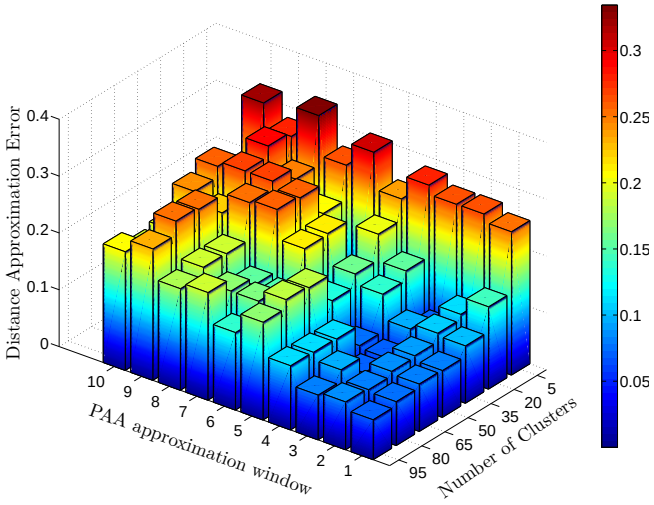


Fig. 3: The trade-off between piece-wise aggregation and symbolic approximation is depicted here comparing the error in approximating the distance between two sequences from the Weizmann dataset. A symbol dictionary size of at least 40 and a approximation window size of up to 3 has negligible approximation error.

ing a distance-based classifier, a nearest-neighbor classifier being the simplest one. When activity sequences involve manifold-valued time-series, distance computations are quite intensive depending on the choice of metrics. We explore here the utility of the symbolic approximation as an alternative way for approximate yet fast recognition of activities that can replace the expensive geodesic distance computations during testing. As we will show in the experiments, this is especially applicable in real-time deployments and in cases where recognition occurs remotely and there is a need to reduce the communication requirements between the sensor and the analysis engine. Before getting into the details of our experiments and distance metrics used, we define some of the terms used in this paper:

1. *Activity* - In this paper, we will consider an activity to be a high dimensional time series consisting of N data points such that each data point is a feature extracted per frame of the original video. The features can be either Euclidean or belong to abstract spaces such as Riemanian manifolds. We consider cases where all activities may not be of equal lengths by using DTW as a distance metric.
2. *Subsequence* - A subsequence is defined as a contiguous subset of the larger time series, i.e. for a time series $T = (t_1, t_2, \ldots, t_n)$ a subsequence of length n is $T_{i,n} = (t_i, t_{i+1}, \ldots, t_{i+n-1})$.
3. *Motif Discovery* - a pattern that repeats often within a larger time series is known as a motif. We say two patterns within the time series are similar if they are at a distance smaller than some threshold.
4. *Trivial Match* - Within a time series T, we say two subsequences P at position p and Q at position q are a trivial match if, $p \in (q-m+1, \ldots, q, \ldots, q+m-1)$ i.e p and q are different and within the neighborhood (as specified by m) of each other.

For an Activity of length N, we extract a symbolic representation in windows of size W (where typically $W << N$). To replace geodesic distance computations for recognition, we will consider subsequences in their symbolic representations to calculate the distance between activities. Let p_{sub} (eg: 'bccdea') and q_{sub} (eg: 'afffec') be two such subsequences of length l, then the distance metric d_{symbol}, defined on symbols, is:

$$d_{symbol}(p_{sub}, q_{sub}) = \sum_{i=1}^{l} d_{\mathcal{M}}\Big(D\big(p_{sub}(i)\big), D\big(q_{sub}(i)\big)\Big) \tag{13}$$

where $d_{\mathcal{M}}$ is the metric defined on the manifold, D is the set of symbols or dictionary that is previously learned and $D(a)$ is the point on the manifold corresponding to the symbol a. Here we assume that the two sequences are of the same length, in other cases we use DTW as a metric or learn a dynamical model for each sequence and use the distance between them as a metric. Since the symbols are known apriori, the distance between them can be computed offline as part of training and stored as a look-up table of pairwise distances between symbols. This allows us to compute distances between sequences in near constant time, which is much faster than computing distances each time using DTW on actual features.

Before considering applications for the simplified distance measure, one must consider the trade-off between piecewise aggregation, number of symbols versus the error of approximation, this is shown in figure 3.

For activity discovery, we consider the problem as one of mining for motifs in time-series. In finding motifs, it is important to consider only non-trivial matches, for every such match we store its location and find the top k motifs. For each of the k motifs, we define a *center* for the motif as the sequence which is at minimum distance to all the sequences similar to it. These centers are the k most recurring patterns in the multidimensional time series. We use the brute-force algorithm given in [Patel et al., 2002] to extract our motifs.

6 Experimental Evaluation

In this section, we demonstrate the utility of the proposed algorithms for symbolic approximation and its application to activity recognition and discovery. We also study the complexity advantage in using these symbols as compared to original feature sequences. We first describe the datasets and

Fig. 4: Sample images from the various data sets used in this paper. The UTKinect [Xia et al., 2012], UMD [Veeraraghavan et al., 2005], the Wiezmann [Gorelick et al., 2007], and the UCSD traffic [Chan and Vasconcelos, 2005] data sets are shown here from top to bottom in that order.

choice of features. Towards this end, we propose a novel approximation method for manifold sequences, which is consistent with the underlying geometry of the manifold, which also lends itself to fast algorithms for sequence indexing, motif discovery etc. among other applications. Solving the manifold valued problem provides us a framework that is general enough to deal with even vector features, wherein they fall under the special case where the manifold is $\mathbb{R}^n$.

We describe the datasets used in this paper next.

UTKinect dataset [Xia et al., 2012] contains 10 activities by 10 subjects, where each activity is repeated twice. There are a total of 199 action sequences. Here we use the feature proposed recently in [Vemulapalli et al., 2014], which models each skeleton as a point on the cross product space of $SE(3) \times \cdots \times SE(3)$.

The UMD database consists of 10 different activities like bend, jog, push, squat etc.[Veeraraghavan et al., 2005], each activity was repeated 10 times, so there were a total of 100 sequences in the dataset. The background within the UMD Dataset is relatively static which allows us to perform background subtraction. From the extracted foreground, we perform morphological operations and extract the outer contour of the human. We sampled a fixed number of points on the outer contour of the silhouette to yield landmarks, which are represented as points on the Grassmann manifold.

The Weizmann Dataset consists of 93 videos of 10 different actions each performed by 9 different persons[Gorelick et al., 2007]. The classes of actions include running, jumping, walking, side walking etc. Here, the HOOF features [Chaudhry et al., 2009] are represented as points on a hyper-spherical manifold.

The UCSD traffic database consists of 254 video sequences of daytime highway traffic in Seattle in three patterns i.e. heavy, medium and light traffic [Chan and Vasconcelos, 2005]. It was collected from a single stationary traffic camera over two days.

Step	Complexity
Exponential map for $\mathcal{M}$ (manifold specific)	O(ν)
Inverse exponential map for $\mathcal{M}$ (manifold specific)	O(χ)
Intrinsic K-means clustering	O((Nχ+Kν) Γ)
Equi-probable clustering	O((NKχ+Nν)Γ)
Approximation of N-length activity to M symbols	O(M(wχ+ν)Γ+MKχ)
Symbolic DTW	O(M$^2\delta$)
Geodesic distance DTW	O(M$^2\chi$), $\chi >>> \delta$

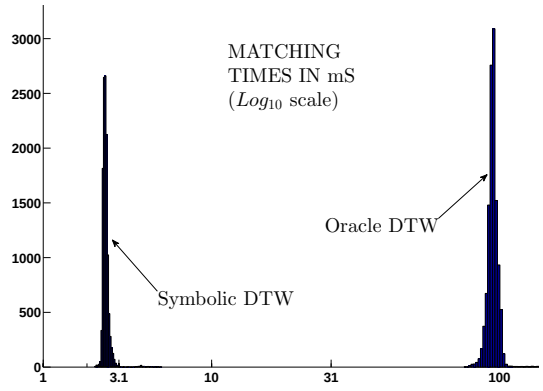
Table 1: Theoretical complexity analysis for the proposed algorithms. Notations used: N - number of data points, K - number of symbols, with O(δ) the time required to read from memory, Γ maximum number of iterations, M and w are as defined in algorithm 2 and are usually much lesser than N. It can be seen that a huge complexity gain is achieved in using symbols over original features.

6.1 Speed up and compression achieved using symbols

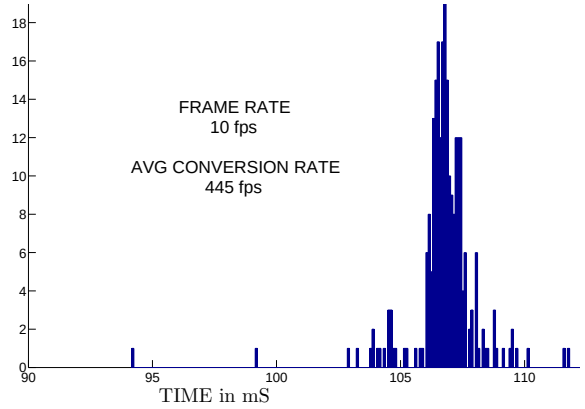
A theoretical complexity analysis of the algorithm is shown in table 1. We also consider three metrics to study the time-complexity of the proposed framework. Namely 1) Time complexity of matching using symbols vs original feature sequences, 2) Time required to transform a given activity into a symbolic form, and 3) Number of bits required to store/transmit symbols as compared to feature sequences. Ideally, we require that the matching time be several orders of magnitude faster than using the original sequences, the transformation time to be small enough to enable real-time approximation, and very small bit-rate/storage requirement compared to original feature sequences. We show in the following that the proposed framework successfully satisfies all these criteria. We performed the experiments using MATLAB, on a PC with an i7 processor operating at 3.40Ghz with 16GB memory on Windows 7.

6.1.1 kNN search and sequence matching time analysis

In this experiment we show the gain in speed and compression achieved using symbols compared to using the original high-dimensional features with accompanying metrics. For the gain in speed, we measured the run-time of matching



(a) Matching times



(b) Converting features to symbols

Manifold	manifold SAX	Geodesic
Grassmann	0.0082 ± 0.0002s	1.54 ± 0.10s
Sphere	0.24 ± 0.08s	1.22 ± 0.39s
$SE(3) \times SE(3)$..	0.50 ± 0.04s	57.80 ± 1.58s

(c) kNN searching times

Feature	Original Data	manifold SAX	Compression
Shape	500 Kb	0.468 Kb	**99.90**%
HOOF feature	65.625 Kb	0.410 Kb	**99.37**%
Skeleton feature	4,617 Kb	111.67 Kb	**97.58**%

(d) Bit budgets for original vs manifold SAX

Fig. 5: Comparison of histograms for matching times when using symbolic v/s original feature sequences are shown in fig 5(a) for the UCSD traffic dataset. The times are shown in milliseconds on a log scale. As it can be seen, using symbols speeds up the process by nearly two orders of magnitude. Fig 5(b) shows a histogram of times taken to translate entire activities of 50 frames into symbols from the UCSD dataset. Table 5(c) shows the improvements in performing a k-NN search on different feature manifolds. Finally table 5(d) shows the reduced storage requirements for different features.

sequences using DTW on symbols vs geodesic DTW. As shown in fig 5(a), the time taken to match two activity sequences using symbols is just 3.1ms which is two orders of magnitude faster than 100ms that it takes using the actual features. Next, we compare the times taken to perform a k-nearest neighbor (kNN) search on different manifolds in table 5(c). Similar to the sequence matching speed, the search speed is improved by nearly two orders of magnitude.

6.1.2 Analysis of approximation time

Fig 5(b) shows the distribution of times taken over various activities to transform them into their respective symbolic forms. The average conversion time for an entire activity video is about 107ms. In other words, we can process the video at a speed of 445 frames per second (fps) which allows for easy real time implementation since most videos are recorded at 10-30fps.

6.1.3 Bit-rate analysis

Next, to demonstrate the gain in compression we compared our representation to a baseline using the original feature sequence. Assuming each dimension of the feature is coded as a 32-bit float number, we calculated the bits it would take to represent each feature and its symbolic representation. As shown in table 5(d), on nearly all the feature types, the compression ratios are 97% or higher. For a dictionary of size K, the number of bits required to represent each symbol is $Log_2(K)$. This provides enough flexibility for the user to choose the size of the dictionary and pick features of their choice without significantly affecting the bit-rate.

6.2 Activity discovery experiment

Having learned the symbols, we test their effectiveness in activity discovery. For this experiment, we randomly concatenated 10 repetitions of 5 different activities of the UMD dataset to create a sequence that was 50 activities long. Each activity consists of 80 frames which were sampled by a sliding window of size 20 frames with step size of 10 frames. After symbolic approximation, this resulted in 6 symbols per activity, chosen from an alphabet of 25 symbols. The *motifs* or repeating patterns, in five activities - *Jogging, Squatting, Bending Knees, Waving and Throwing* were discovered automatically using the proposed method. Each of the discovered motifs was validated manually to obtain a confusion matrix shown in table 2. As can be seen, it shows a strong diagonal structure, which indicates that the algorithm works fairly well. Even though all executions of the same activity are not found, we do not find any false matches either.

Activity Type	1	2	3	4	5
1	7	0	0	0	0
2	0	7	0	0	0
3	0	0	8	0	0
4	0	0	0	9	0
5	0	0	0	0	8

Table 2: Confusion matrix for the discovered motifs on the UMD database using the manifold SAX representation of the shape feature. Due to the symbolic representation, search can be performed very quickly. Actions discovered are - *jogging, squatting, bending, waving and throwing* respectively.

6.3 Activity recognition using symbols

Symbolic approximation plays a significant role in reducing computational complexity since it allows us to work with symbols instead of working with high dimensional feature sets. In this experiment, we test the utility of the proposed symbolic approximation method for fast and approximate recognition of activities over three datasets. For each data set picking the number of symbols, K is an empirical choice, typically we picked $K = K_{min}$ where, for all $K > K_{min}$ the recognition performance shows no improvement. We also picked a window size of $W = 1$ in our recognition experiments to achieve best performance. A detailed comparison between the window size, number of symbols and performance is seen in figure 3, which shows the the error in the geodesic distance vs symbolic distance. To effectively demonstrate the quality of the approximation, we use the classifiers that were reported in the papers that proposed the features. For example, for the shape and the HOOF features, we use the nearest neighbor classifiers, and for the LARP features, we use the SVM.

Activity	Accuracy	Relative bit budget
Shape + manifold SAX	98	1
Shape + PGA [Fletcher et al., 2004]	90	6.012
Shape [Veeraraghavan et al., 2005]	100	1202.6

Table 3: Recognition experiment for the UMD database with a shape silhouette feature. Here we see the performance achieved with symbolic approximation compared to an oracle geodesic distance based nearest neighbor classifier.

For the UMD dataset, we learned a dictionary of 60 symbols using algorithm 1. Then, we performed a recognition experiment using a leave one-execution-out test in which we trained on 9 executions and tested on the remaining execution, the results are shown in Table 3. It can be seen that the recognition performance using symbols is very close to that obtained by using an oracle geodesic distance DTW based algorithm. We achieve this performance with matching times that are significantly faster, as will be described in section 6.1.

For the UTKinect dataset, we learn a common alphabet of size $20-25$ symbols for all the relative joints from actions corresponding to the training subjects. The approximated LARP features are then mapped to their corresponding Lie algebra following the protocol of [Vemulapalli et al., 2014]. Finally these features are classified using a one-vs-all SVM classifier similar to [Vemulapalli et al., 2014]. Results show that even with a small dictionary size, there is negligible loss in recognition accuracy, while drastically reducing the search speed 5(c) by a factor of nearly 50. We also compare the recognition accuracy of principal geodesic analysis (PGA) [Fletcher et al., 2004] on the Lie algebra.

Feature	Accuracy	Relative bit budget
LARP+ manifold SAX	**94.77**	1
LARP+PGA [Fletcher et al., 2004]	92.46	20.428
LARP [Vemulapalli et al., 2014]	92.97	40.856
HOG3D [Xia et al., 2012]	90.00	NA

Table 4: Results on the UTKinect dataset.

For the Weizmann dataset, we demonstrate the flexibility of the approximation strategy by learning linear dynamical models over the approximated sequences, which also serves as a fair comparison to the state of the art techniques. We performed the recognition experiment on all the 9 subjects performing 10 activities each with a total of 90 activities. The dictionary learned had 55 symbols which were used to map the activities to the approximated sequences. Next, we fit a linear dynamical model to the approximately reconstructed actions and perform recognition with a nearest neighbor classifier using the Martin metric on LDS parameters [Soatto et al., 2001]. The results for the leave-one-execution-out recognition test are shown in Table 5 and it can be seen there is almost no loss in performance in comparison to state of the art techniques. Better results have been reported on this dataset by Gorelick *et al.* [Gorelick et al., 2007] etc., but there are no common grounds between their technique or feature and ours for it to be a fair comparison.

For the Traffic Database, we stacked every other pixel in the rows and columns of each frame to form our feature vector. We learned 45 symbols from the training set using these features. We performed the recognition exper-

LDS+ manifold SAX	**92.22**
HOOF+DTW+manifold SAX	88.87
HOOF+DTW [Chaudhry et al., 2009]	90.00
χ^2 Kernel[Chaudhry et al., 2009]	95.66
HIST Kernel[Chaudhry et al., 2009]	92.33
Chaotic measures[Ali et al., 2007]	92.60

Table 5: Recognition Performance for the Weizmann dataset.

	Manifold SAX	CS LDS	Oracle LDS
Expt 1	84.13	85.71	77.77
Expt 2	82.81	73.43	82.81
Expt 3	79.69	78.10	91.18
Expt 4	79.37	76.10	80.95
Average	**81.50**	78.33	83.25

Table 6: Recognition performance for UCSD traffic data set. The results for Oracle LDS and CS LDS are from [Sankaranarayanan et al., 2010].

iment on 4 different test sets which contained 25% of the total videos. We used a 1-NN classifier with a DTW metric on the symbols. The results are shown in Table 6. We compare our results to [Sankaranarayanan et al., 2010], which also performed recognition using lower dimensional feature representation using compressive sensing. As it can be seen, recognition performance is clearly better when the feature is in its symbolic form as compared to when it was compressively sensed, given that both are significantly reduced versions of the original feature. We also perform nearly as well as the performance achieved using the original feature itself.

7 Discussion and Future Work

In this paper we presented a formalization of high dimensional time-series approximation for efficient and low-complexity activity discovery and activity recognition. We presented geometry and data adaptive strategies for symbolic approximation, which enables these techniques for new classes of non-Euclidean visual representations, for instance in activity analysis. The results show that it is possible to significantly reduce Riemannian computations during run-time by an intrinsic indexing and approximation algorithm which allows for easy and efficient real time implementation. This opens several avenues for future work like an integrated approach of temporal segmentation of human activities and symbolic approximation. A theoretical and empirical analysis of the advantages of the proposed formalism on resource-constrained systems such as robotic platforms would be another avenue of research.

Finally, the framework in this paper is general enough to deal with more abstract forms of information such as graphs [Jordan, 1998] or bag-of-words [Gaur et al., 2011]. In fact, any system that is sequential can be used within this framework, the key is to have a good understanding of metrics on these abstract models. Existing works have defined kernels for data on manifolds [Lafferty and Lebanon, 2005], for graphs [Vishwanathan et al., 2008] and a good starting point would be to use these to develop a kernel version of this framework that would allow us to learn symbols.

References

Absil et al., 2004. Absil, P.-A., Mahony, R., and Sepulchre, R. (2004). Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220. 3

Ali et al., 2007. Ali, S., Basharat, A., and Shah, M. (2007). Chaotic invariants for human action recognition. In *ICCV*, pages 1–8. 12

Allauzen and Raffinot, 2000. Allauzen, C. and Raffinot, M. (2000). Simple optimal string matching algorithm. In *Combinatorial Pattern Matching*, volume 1848 of *Lecture Notes in Computer Science*, pages 364–374. Springer Berlin Heidelberg. 3, 5

Boothby, 2003. Boothby, W. M. (2003). *An Introduction to Differentiable Manifolds and Riemannian Geometry. Revised 2nd Ed.* Academic, New York. 3

Chakrabarti et al., 2002. Chakrabarti, K., Keogh, E. J., Mehrotra, S., and Pazzani, M. J. (2002). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Database Syst.*, 27(2):188–228. 5

Chan and Vasconcelos, 2005. Chan, A. and Vasconcelos, N. (2005). Classification and retrieval of traffic video using auto-regressive stochastic processes. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 771–776. 9

Chaudhry and Ivanov, 2010. Chaudhry, R. and Ivanov, Y. (2010). Fast approximate nearest neighbor methods for non-Euclidean manifolds with applications to human activity analysis in videos. In *European Conference on Computer Vision*, Crete, Greece. 2

Chaudhry et al., 2009. Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR, 2009.*, pages 1932 –1939. 1, 4, 9, 12

Chum et al., 2009. Chum, O., Perdoch, M., and Matas, J. (2009). Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, pages 17–24. 3

Desieno, 1988. Desieno, D. (1988). Adding a conscience to competitive learning. *IEEE International Conference on Neural Networks*, 1:117–124. 2, 5

Devroye et al., 1992. Devroye, L., Szpankowski, W., and Rais, B. (1992). A note on the height of suffix trees. *SIAM Journal on Computing*, 21(1):48–53. 3, 5

Fletcher et al., 2004. Fletcher, P. T., Lu, C., Pizer, S. M., and Joshi, S. C. (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005. 11

Gaur et al., 2011. Gaur, U., Zhu, Y., Song, B., and Chowdhury, A. K. R. (2011). A "string of feature graphs" model for recognition of complex activities in natural videos. In *ICCV*, pages 2595–2602. 12

Goodall and Mardia, 1999. Goodall, C. R. and Mardia, K. V. (1999). Projective shape analysis. *Journal of Computational and Graphical Statistics*, 8(2). 3

Gorelick et al., 2007. Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2247 –2253. 9, 11

Harandi et al., 2014. Harandi, M. T., Salzmann, M., and Hartley, R. (2014). From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices. In *ECCV*, pages 17–32. 2

Jordan, 1998. Jordan, M. I. (1998). *Learning in Graphical Models*. Cambridge, MA: MIT Press. 12

Joshi et al., 2007. Joshi, S. H., Klassen, E., Srivastava, A., and Jermyn, I. (2007). A novel representation for Riemannian analysis of elastic curves in R^n. In *CVPR*. 1, 2

Karcher, 1977. Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541. 5

Kendall, 1984. Kendall, D. (1984). Shape manifolds, Procrustean metrics and complex projective spaces. *Bulletin of London Mathematical society*, 16:81–121. 1

Kohonen, 1995. Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer - Verlag. 5

Lafferty and Lebanon, 2005. Lafferty, J. D. and Lebanon, G. (2005). Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163. 12

Lin et al., 2003. Lin, J., Keogh, E. J., Lonardi, S., and chi Chiu, B. Y. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *DMKD*, pages 2–11. 2, 3, 5

Lin and Li, 2010. Lin, J. and Li, Y. (2010). Finding approximate frequent patterns in streaming medical data. In *CBMS*, pages 13–18. 5

Lin and Zha, 2008. Lin, T. and Zha, H. (2008). Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:796–809. 7

Lui et al., 2010. Lui, Y. M., Beveridge, J. R., and Kirby, M. (2010). Action classification on product manifolds. In *CVPR*, pages 833–839. 1

Mueen et al., 2009. Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., and Westover, M. B. (2009). Exact discovery of time series motifs. In *SDM*, pages 473–484. 3

Patel et al., 2002. Patel, P., Keogh, E., Lin, J., and Lonardi, S. (2002). Mining motifs in massive time series databases. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 370–377. 8

Pennec, 2006. Pennec, X. (2006). Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154. 5

Pennec et al., 2006. Pennec, X., Fillard, P., and Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66. 1

Rahman et al., 2005. Rahman, I. U., Drori, I., Stodden, V. C., Donoho, D. L., and Schrder, P. (2005). Multiscale representations for manifold-valued data. *SIAM J. MULTISCALE MODEL. SIMUL*, 4(4):1201–1232. 2

Revaud et al., 2013. Revaud, J., Douze, M., Schmid, C., and Jegou, H. (2013). Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, pages 2459–2466. 3

Ripley, 1996. Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press. 5

Roweis and Saul, 2000. Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326. 7

Sankaranarayanan et al., 2010. Sankaranarayanan, A. C., Turaga, P. K., Baraniuk, R. G., and Chellappa, R. (2010). Compressive acquisition of dynamic scenes. In *ECCV (1)*, pages 129–142. 11, 12

Soatto et al., 2001. Soatto, S., Doretto, G., and Wu, Y. N. (2001). Dynamic textures. *ICCV*, 2:439–446. 11

Spivak, 1999. Spivak, M. (1999). *A Comprehensive Introduction to Differential Geometry*, volume One. Publish or Perish, Inc., Houston, Texas, third edition. 3

Srivastava et al., 2007. Srivastava, A., Jermyn, I., and Joshi, S. (2007). Riemannian analysis of probability density functions with applications in vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. 4

Srivastava et al., 2011. Srivastava, A., Klassen, E., Joshi, S. H., and Jermyn, I. H. (2011). Shape analysis of elastic curves in Euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1415–1428. 1, 2, 3

Su et al., 2014. Su, J., Kurtek, S., Klassen, E., and Srivastava, A. (2014). Statistical analysis of trajectories on Riemannian manifolds: Bird migration, hurricane tracking, and video surveillance. *Annals of Applied Statistics*, 8(1). 2

Tenenbaum et al., 2000. Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323. 7

Turaga et al., 2010. Turaga, P., Veeraraghavan, A., Srivastava, A., and Chellappa, R. (2010). Statistical analysis on manifolds and its applications to video analysis. In Schonfeld, D., Shan, C., Tao, D., and Wang, L., editors, *Video Search and Mining*, volume 287 of *Studies in Computational Intelligence*, pages 115–144. Springer Berlin Heidelberg. 4

Turaga and Chellappa, 2009. Turaga, P. K. and Chellappa, R. (2009). Locally time-invariant models of human activities using trajectories on the Grassmannian. In *CVPR*, pages 2435–2441. 1

Turaga et al., 2011. Turaga, P. K., Veeraraghavan, A., Srivastava, A., and Chellappa, R. (2011). Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286. 1, 3

Tuzel et al., 2006. Tuzel, O., Porikli, F. M., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. *European Conference on Computer Vision*, pages II: 589–600. 1

Vahdatpour et al., 2009. Vahdatpour, A., Amini, N., and Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *IJCAI*, pages 1261–1266. 3

Veeraraghavan and Chowdhury, 2006. Veeraraghavan, A. and Chowdhury, A. K. R. (2006). The function space of an activity. In *CVPR (1)*, pages 959–968. 6

Veeraraghavan et al., 2005. Veeraraghavan, A., Chowdhury, A. K. R., and Chellappa, R. (2005). Matching shape sequences in video with applications in human movement analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1896–1909. 9, 11

Vemulapalli et al., 2014. Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). Human action recognition by representing 3d skeletons as points in a lie group. In *(CVPR), 2014*, pages 588–595. 1, 4, 9, 11

Vishwanathan et al., 2008. Vishwanathan, S. V. N., Borgwardt, K. M., Kondor, I. R., and Schraudolph, N. N. (2008). Graph kernels. *CoRR*, abs/0807.0093. 12

Xia et al., 2012. Xia, L., Chen, C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW)2012*, pages 20–27. IEEE. 9, 11

Yao, 1979. Yao, A. (1979). The complexity of pattern matching for a random string. *SIAM Journal on Computing*, 8(3):368–387. 3

Yi et al., 2012. Yi, S., Krim, H., and Norris, L. K. (2012). Human activity as a manifold-valued random process. *IEEE Transactions on Image Processing*, 21(8):3416–3428. 2

Zador, 1982. Zador, P. (1982). Asymptotic quantization error of continuous signals and the quantization dimension. *Information Theory, IEEE Transactions on*, 28(2):139 – 149. 5